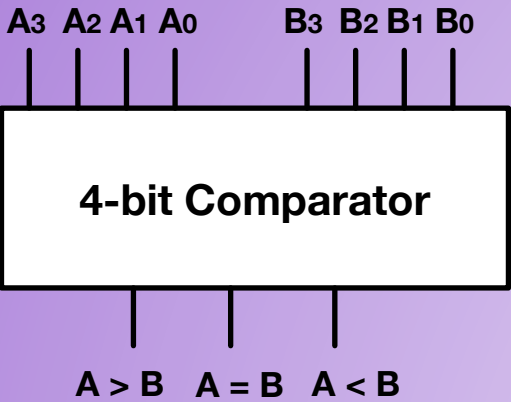
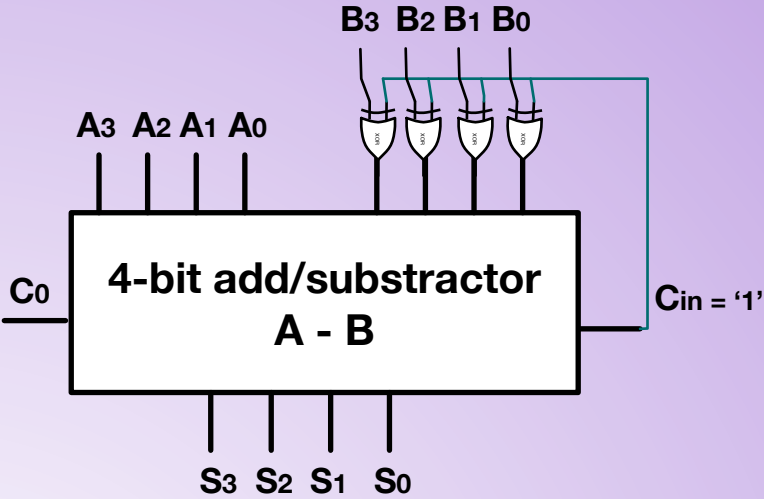


Comparator



COMBINATIONAL
COMPARATOR



Design from scratch

Using Subtractor

All examples are for 4-bit comparator, similar approach is used for larger size

A = B $E_0 + E_1 + E_2 + E_3$

A > B $A_3.B'_3 + E_3.A_2.B'_2 + E_3.E_2.A_1.B'_1 + E_3.E_2.E_1.A_0.B'_0$

A < B $A'_3.B_3 + E_3.A'_2.B_2 + E_3.E_2.A'_1.B_1 + E_3.E_2.E_1.A'_0.B_0$

E_n checking equality of bit i $E_n = A_n.B_n + A'_n.B'_n$

$E_n = 1$ two bits A_n and B_n are equal

$A_0 = B_0 \Rightarrow A_0 = B_0 = 1 \text{ OR } A_0 = B_0 = 0$

$\Rightarrow E_0 = A_0.B_0 + A'_0.B'_0$

$\Rightarrow E_0 = A_0 \text{ XNOR } B_0$

Checking inequality

A > B if (A3 > B3)

OR ((A3 = B3) and (A2 > B2))

OR ((A3 = B3) and (A2 = B2) and (A1 > B1))

OR ((A3 = B3) and (A2 = B2) and (A1 = B1) and (A0 > B0))

A = B Cout = '1' and $S_3 = S_2 = S_1 = S_0 = 0$

A > B Cout = '1' and one of $\{S_3, S_2, S_1, S_0\} = 1$

A < B Cout = '0'

Observation

A = 5 (0101) B = 9 (1001) A - B = 0101 0111 0 1100	A = 5 (0101) B = 5 (0101) A - B = 0101 1011 1 0000	A = 5 (0101) B = 4 (0100) A - B = 0101 1100 1 0001
---	---	---

Carry out

